

SOFTWARE MAINTENANCE AWARE TEST CASE GENERATION TECHNIQUE

G.Krishnan

Coordinator, Admissions
Krishnan.g@srmuniv.ac.in
SRM University, SRM Nagar, Kattankulathur-603203.

ABSTRACT

The software maintenance is a major and high cost consuming task. This paper identifies and describes a new approach of software maintenance aware test case generation techniques which makes the simpler and cost effective. Software is tested by creating test case, created test case is tested by giving as an input to the software system and the resultant is compared to validate the test case. The creating the test case and test data extraction plays a major role in validating the software system and during the software maintenance which takes few modules added or updated the pre formulated test case will not able to use. This paper attempts to create a test case generation technique which is optimal and usable even few modules are added and updated. More research are focused towards optimizing and reducing test cases this also going to do so but only difference is software

maintenance is focused during the optimizing.

Keywords: *Software maintenance, test case generation, optimizing techniques*

1. INTRODUCTION

Software testing and software maintenance are the two main areas in software development life cycle. After the software project is delivered to the customer, it is very much needed to maintain the software project for a long time process. Software maintenance activity consumes a major portion of the total life cycle cost of a system. Creating test case is also an important activity in testing phase this paper focuses to creating a maintenance aware test case creation. Since change is inevitable, new methods must be developed for evaluating, controlling and making modifications. The main task in this paper is software maintenance which remains an opaque activity that is expensive and difficult to manage. Software Maintenance activities can be divided into two types one is modification and other is debugging. Modification, when there is a need to change according to the environment. Debugging, when there is a need to correct faults, this is happened earlier. This Paper contains to develop software

maintenance aware test case generation with taxonomies of software maintenance activities. Change request form is main source of data collection. Analysis has been reviewed and categorized into different types of software maintenance.

2. FACTORS OF THE SOFTWARE MAINTENANCE BASED TEST CASE CREATION ACTIVITY

The set of factors which considered during the creation of software maintenance based test case creation activity are to provide good service which means fixing bugs, recovering from failures and accommodating changes in the operating system and hardware. To hold upgrades which means changes in government regulations or to maintain a competitive product To carry user requests for improvements which means enhancement of functionality, better performance and customization to local working patterns. To make easy future maintenance work this means code and database restructuring and updating documentation.

3. SOFTWARE MAINTENANCE AWARE TESTCASE GENERATION TECHNIQUE

To compass up with the quick changing and demanding in software development, we need to modify our software in its maintenance phase. This section describes the Software maintenance aware test case generation technique. The maintenance aware test case is generated

through the test cases require maintenance aware test data for their execution which makes the test data generation a building block activity in the overall test cases execution process. In usual process test data can be resultant from different UML models as well as other different types of models. Search based testing models are one of the most important models used; they include evolutionary models and Genetic algorithms. In order to be able to claim that the maintenance aware test cases are better than others or even decide whether they are applicable or not; they must be first qualified for usage. Quality of test cases depends on how well they cover the functionalities of the system under test and not only on their form. After modifications are made, we need to retest the software using existing test suite so that we regain the confidence in correctness of our system. This is called maintenance aware testing. Maintenance test suites being too large to re-execute in the given time and cost constraints are reduced or re-ordered. This can be achieved by using one or more of the three techniques, Test case selection, minimization or prioritization. Maintenance aware test case is the reordering of the test suite according to an appropriate criterion like code, branch, condition and fault coverage etc. We can also select a subset of the original test suite on the basis of some criteria, often called as Regression Test Selection or using Test case minimization we can identify and remove the redundant test cases. Considering the cost of executing a test case, many cost-aware maintenance techniques have been proposed Taking time as cost, Time-

Aware test suite maintenance was proposed. It uses execution time of the test cases as a parameter for maintenance aware test case in addition to fault identification criteria. Execution time acts as the cost of executing the test case. Maintenance aware test cases are then done according to maximum fault identification and minimum cost of execution. Time constrained test case prioritization problem has been reduced to zero/one knapsack problem which is NP-complete. Thus, techniques that solve combinatorial optimization problems can be applied to time constrained maintenance aware test cases. The test cases should be validated against predefined quality standards which determine their acceptable form as well as the degree of their functional coverage which in turn specifies their level of applicability. Many metrics have emerged and are being used to measure the quality of the test cases being generated like the time, cost, effort, and complexity of generation, coverage criteria and many others. Optimizing or even improving the quality of the test cases can be the aim of some researchers. It can take several forms, such as decreasing the testing effort or time, decreasing the complexity or cost of the generation algorithms, increasing the functionality coverage as well as other quality and reliability issues. Also reducing the generated test cases or test data can be a form of optimization. Many tools and frameworks have been implemented; some are used for maintenance aware test cases generation, others used for test data extraction, whereas others are for reducing already existing sets of test cases or test data by

selecting subsets from them, trying by this to optimize the use of test cases or test data.

4. CONCLUSION

In the present scenario, consumers have become more aware about product quality. In response, manufacturers are changing their company attitude. Customer satisfaction and nonstop development of product quality have become the objectives. Companies identify that it's their responsibility to determine if a proposed change could significantly affect security or efficiency of a product. Manufacturers realize that effective change control is essential to continuous quality improvement, which can increase customer satisfaction. The change management is only found in the software industry background, and getting the company data is very difficult. With the cooperation of the maintenance and test case creation is needed, data has been collected. This paper attempted to create a test case generation technique which is optimal and usable even few modules are added and updated. This research are focused towards optimizing and reducing test cases this also going to do so but only difference is software maintenance is focused during the optimizing.

REFERENCES

- [1] S.Y. Stephen 1978, "Ripple Effect Analysis of software maintenance", in Proc.COMPSAC78.
- [2] B.P. Lientz and E.B.Swanson (1980), "Software Maintenance Management".

- Addison Wesley, Reading, MA. [Kas09] geographically Distributed
Kaseem A.Saleh (2009), “Software Environments”.
- [3] Rogers S.Pressman (2005), “Software Engineering: A Practitioner’s Approach “Sixth Edition, McGraw-Hill.
- [12] Gopalaswamy Ramesh (2002) “Managing Global Software Projects”. “CM014F” Data sheet, 3I InfoTech, Chennai. “
- [4] Ian Sommerville (1996), “Software Engineering”, Fifth Edition. PP 259-262.
- [5] IEEE Std 1219-1998, “IEEE Standard for Software Maintenance”. IEEE, 1998.
- [6] IEEE Standard for Software Configuration management. (1998). IEEE std 1219 -1998 Software Engineering Standards Committee of the IEEE Computer society.
- [7] T.M. Pigoski, “Practical Software Maintenance: Best Practices for Managing your Software Investment”, first ed., John Wiley & Sons, 1997.
- [8] B.W. Boehm (1981), “Software Engineering Economics”, Prentice Hall.
- [9] K.K Aggarval and Yogesh Singh (2005), “Software engineering: Programs, Documentation, and Operating Procedures”. Second edition.
- [10] Bee Bee Chual and June Verner “Examining Requirements Change Rework Effort: A Study “, International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.3, July 2010.
- [11] Gopalaswamy Ramesh & Ramesh Bhattiprolu (2006) “Software Maintenance Effective Practices for